

Using SAS for Variance Estimation with Complex Sample Surveys

Patricia A. Berglund

Institute For Social Research-University of Michigan

Section 1-Background Information on Complex Sample Surveys

Complex Surveys and Simple Random Samples

Complex surveys are based on sample designs which involve clustering, stratification, and/or weights which adjust for non-response and differing probabilities of selection. Complex samples differ from simple random samples (SRS) in that simple random samples assume independence of observations or a "simple" sample design while complex samples do not. The programs presented in this paper were written for use with the National Comorbidity Survey (NCS) Dr. Ronald Kessler-Principal Investigator. The NCS will be used throughout as an example of a complex survey.

The NCS is a nationally representative sample based on a stratified, multi-stage area probability sample of the United States.¹ Weights which adjust for non-response and differing probabilities of selection are routinely used in analyses. The NCS data file also includes 2 key variables which allow analysts to incorporate the complex survey design into variance estimation computations, (Kalton, 1977). These 2 variables are called the stratum and secu (Sampling Error Computing Unit). The NCS stratum codes range from 1 to 42 and reflect geographic areas sampled. For example, self-representing areas of the country are represented by a single stratum while non-self representing areas are combined into one stratum. Three very large cities, New York, Los Angeles, and Chicago are split into 2 strata.

The second key variable, the secu, is a stratum-specific half sample code for analysis of sampling error using the BRR (Balanced Repeated Replication) method or approximate "two-per-stratum" Taylor Series method (Kish and Hess, 1959).² These codes can also be used for JRR (Jackknife Repeated Replication). Secu codes for self-representing strata are done randomly and are designed to preserve the stratification and second stage clustering properties of the sample within a self-representing stratum.³ For non self-representing strata, half sample codes are defined according to the primary sampling unit assigned at sample selection.⁴ With the creation of the stratum and secu, the complexity of the sample design is accounted for and corrected variance estimates can be calculated by using one of two methods.

Why Re-calculate Variance Estimates for Complex Samples?

Statistics produced by standard packages such as SAS and SPSS assume a Simple Random Sample and result in under-estimation of variances for all statistics based on complex samples. This is due to the nature of the complex design. When compared to SRS, complex designs cause larger standard errors because of clustering and weighting while decreased standard errors are a result of stratification. The combined effect of these processes is that standard error estimates from SAS programs that assume independent sampling of observations under-estimate the true variance of the complex sample. Therefore, analyses of data from complex surveys should include special calculation of variance estimates which account for these sample characteristics. In general, these corrections are performed at the end of the analysis process, once the final models are developed and more precise estimates are needed.

Common Methods of Adjusting Variance Estimates for Design Effects

The question of how to correctly calculate standard errors and confidence intervals for complex samples can be answered in a variety of ways. The methods available to analysts can be divided into 2 main groups: linearization and resampling techniques.

The Taylor Series Linearization Method

The Taylor Series Linearization Approach (Kish, 1965) is based on a method which derives a linear approximation of variance estimates which are used to develop corrected standard errors and confidence intervals for descriptive statistics such as means and proportions as well as regression coefficients. A major advantage of the Taylor Series is that it is very efficient computationally since replicate models do not have to be calculated. For example, a high end Pentium PC can quickly handle small and medium size datasets (up to 8,000 records) and reasonable regression models (up to 20 predictors.) Commercially available software such as SUDAAN, STATA, and PC CARP use the Taylor Series Method.

Resampling Approaches

Resampling approaches follow a specified method of selecting replicates which are probability sub-samples, or replicates from which variance estimates are derived. Because the formulation of the probability samples is based upon the complex design, unbiased, design-corrected variance estimates can be developed.

Commonly used methods of resampling include Balanced Repeated Replication and Jackknife Repeated Replication (JRR), (Wolter, 1985). Balanced Repeated Replication (BRR) is a method which reduces the number of sub-samples needed by dividing each stratum into halves. This is accomplished by using the secu codes which have values of 1 or 2. Once the statistic of interest is derived from the half samples (or replicates) the design corrected variance estimations can be developed.

NCS analysts generally prefer the Jackknife Repeated Replication method because it provides the largest replicate sample size with a reasonable number of iterations and is relatively easy to program in SAS or alternate software. The process is to delete a small and different portion of the total sample, run the statistics of interest on each replicate, and repeat this process a prescribed number of times. After completing the total sample and replicate runs, variance estimates can then be calculated. This process will be explained in detail in the next section.

Section 2-Presentation of the SAS Program

Demonstration of the Jackknife Approach for Standard Errors for Logistic Regression Parameters

This section presents in detail the JRR logistic regression program which calculates design corrected standard errors for regression parameters.

Part 1: Creation of Outer Macro and Jackknife Replicates

Jackknife for logistic regression, written by Pat Berglund during winter 97;
Uses National Comorbidity Survey Data, Dr. Ronald Kessler, director**;

```
1 libname max 'c:\data';
2 %macro jacklog(x1,x2,x3,x4,x5,x6,x7,numclus,numpred);
3 data jrr;
4   set max.ncsdxdm2;
5   p1=&x1;
6   p2=&x2;
7   p3=&x3;
8   p4=&x4;
9   p5=&x5;
10  p6=&x6;
11  p7=&x7;

12 %macro wgtcal;
13   %do i=1 %to &numclus;
14     partwt&i=p1fwt;
15     if str=&i and secu=1 then partwt&i=partwt&i*2;
16     if str=&i and secu=2 then partwt&i=0;
17   %end;

18 %mend;
19 %wgtcal;
```

Part 1 defines the outer macro for entire program, macro "jacklog", and then sets up the actual Jackknife process in macro "wgtcal" (lines 12-19). Macro "wgtcal" uses 42 strata times 2 secu values for a total of 84 possible strata/secu combinations. "Wgtcal" iterates 42 times and for each iteration takes 1/84th of the total and multiplies by 2 while zeroing out the complementary strata/secu combination. For example, iteration 1 multiplies str=1/ secu=1 by 2 and zeros out str=1/secu=2. All other strata/secu combinations will remain the same. Iteration 2 would multiply str=2/secu=1 by 2 and set str=2/secu=2 to zero and all other combinations are the same. This process is repeated 42 times, forming the 42 replicates.

Part 2: Run Base and Replicate Procedures

```

*create the base predictors ;
20 proc logistic des data=jrr outest=parms ;
21   model depl1=p1-p7 ;
22   weight p1fw / norm ;
23   run ;

24 %macro reps ;
25 %do i=1 %to &numclus ;
26   proc logistic des data=jrr outest=parms&i noprint ;
27     model depl1=p1-p7 ;
28     weight partwt&i / norm ;
29     run ;
30     data parms&i ;
31     set parms&i ;
32     %do j=1 %to &numpred ;
33       rename p&j=p&j.x&i ;
34     %end ;
35 %end ;
36 %mend ;
37 %reps ;

```

Part 2 runs the "base" model for the entire sample (lines 20-23) and then runs the same model 42 times for each of the replicates developed in part 1 (lines 24-37 or macro "reps"). With the base estimates and 42 sets of replicate estimates, we can then calculate the correct variance and standard errors for parameters, based on the sample design.

Part 3: Calculation of Variance and Standard Errors

```

38 data rep ;
39   merge
      parms
      parms1 parms2 parms3 parms4 parms5 parms6 parms7 parms8 parms9 parms10
      parms11 parms12 parms13 parms14 parms15 parms16 parms17 parms18 parms19 parms20
      parms21 parms22 parms23 parms24 parms25 parms26 parms27 parms28 parms29 parms30
      parms31 parms32 parms33 parms34 parms35 parms36 parms37 parms38 parms39 parms40
      parms41 parms42 ;
40   run ;
41   data all ;
42   set rep ;
43   %macro it ;
44     %do j=1 %to &numpred ;
45       %do i=1 %to &numclus ;
46         dp&j.x&i=(p&j-p&j.x&i)**2 ;
47       %end ;
48     %end ;
49   %end ;
50   %macro final ;
51     %do j=1 %to &numpred ;
52       data last ;
53       set all ;
54       sp&j=sum(of dp&j.x1-dp&j.x42) ;
55       sep&j=sqrt(sp&j) ;
56       orp&j=exp(p&j) ;
57       lp&j=exp(p&j-(1.96*sep&j)) ;
58       up&j=exp(p&j+(1.96*sep&j)) ;
59       proc print ;
60         var p&j sep&j orp&j lp&j up&j ;
61       run ;
62     %end ;
63 %mend ;
64 %final ;

```

```
65 %mend ;  
66 %jacklog(sexf,ed011,ed12,ed1315,r4mw,r4ne,r4s,42,7) ;
```

Part 3 merges the base and replicate temporary datasets together (lines 38-39), creates 42 variables which are the squares of the differences between the base parameter estimate and the 42 replicate parameters (macro "it"), and then finally calculates the variance (line 54) which is the sum of the 42 squared differences, the standard error which is the square root of the variance (line 55), the odds ratio (exponent of the parameter estimate, line 56), and the confidence interval of the odds ratio (lines 57-58).

Summary

Two additional programs are included in the appendix. Both use the Jackknife procedure for estimating corrected variances for means (Appendix program 1) and standard errors of parameters from Ordinary Least Squares regression (Appendix program 2). The process of developing the replicates and calculating the variance estimates do not differ; only the procedures change. These programs can be easily modified to accommodate other complex samples with any number of stratum as well as varying numbers of parameters or means. They provide a convenient way to use SAS and the SAS Macro language to efficiently calculate design based variance estimates.

References

- Heeringa, S. (1997). "International Consortium in Psychiatric Epidemiology (ICPE): "Sampling error Estimation and Inference for Complex Designs". Ann Arbor: Institute for Social Research
- Heeringa, S. (1996). "National Comorbidity Survey (NCS): Procedures for Sampling Error Estimation". Ann Arbor: Institute for Social Research
- Hill, M.S. (1992). "The Panel Study of Income Dynamics". Sage Publications, London. England
- Kalton, G. (1977). "Practical Methods for Estimating Survey Sampling Errors," Bulletin of the International Statistical Institute, Vol 47, 3, pp. 495-514.
- Kish, L. and Frankel, M.R. (1974). "Inference from complex samples," Journal of the Royal Statistical Society, B, Vol. 36, pp. 1-37.
- Kish, L., (1965). Survey Sampling. John Wiley and Sons, Inc., New York.
- Kish, L., and Hess, I. (1959). "On variances of ratios and their differences in multi-stage samples," Journal of the American Statistical Association, 54, pp. 416-46.
- Wolter, K.M. (1985). Introduction to Variance Estimation. New York: Springer-Verlag.

Footnotes

1. From Heeringa, S. "National Comorbidity Survey (NCS): Procedures for Sampling Error Estimation".
2. From Heeringa, S. "National Comorbidity Survey (NCS): Procedures for Sampling Error Estimation".
3. From Heeringa, S. "National Comorbidity Survey (NCS): Procedures for Sampling Error Estimation".
4. From Heeringa, S. "National Comorbidity Survey (NCS): Procedures for Sampling Error Estimation".

Appendix Program 1

```

***Jackknife for means , written by Pat Berglund during winter 97**** ;
***Uses National Comorbidity Survey Data, Dr. Ronald Kessler, director***** ;
libname max 'c:\data' ;
options symbolgen mprint ;

%macro jackmean(x1,x2,x3,x4,x5,numclus,numvar) ;

data jrr ;
    set max.ncsdxdm2;
    p1=&x1 ;
    p2=&x2 ;
    p3=&x3 ;
    p4=&x4 ;
    p5=&x5 ;
%macro wgtcal ;
    %do i=1 %to &numclus ;
        partwt&i=p1fwt ;
        if str=&i and secu=1 then partwt&i=partwt&i*2 ;
        if str=&i and secu=2 then partwt&i=0 ;
    %end ;

%mend ;
%wgtcal ;

*create the base predictors ;
proc means data=jrr ;
    var p1-p5 ;
    weight plfwt ;
    output out=base mean=m1 m2 m3 m4 m5 ;
run ;

%macro reps ;
    %do i=1 %to &numclus ;
        proc means data=jrr noprint ;
            var p1-p5 ;
            weight partwt&i ;
            output out=rep&i mean=m1r&i m2r&i m3r&i m4r&i m5r&i ;
        run ;
    %end ;
%mend ;
%reps ;

data rep ;
    merge
        base
        rep1 rep2 rep3 rep4 rep5 rep6 rep7 rep8 rep9 rep10
        rep11 rep12 rep13 rep14 rep15 rep16 rep17 rep18 rep19 rep20
        rep21 rep22 rep23 rep24 rep25 rep26 rep27 rep28 rep29 rep30
        rep31 rep32 rep33 rep34 rep35 rep36 rep37 rep38 rep39 rep40
        rep41 rep42 ;
run ;
data all ;
    set rep ;
%macro it ;
    %do j=1 %to &numvar ;
        %do i=1 %to &numclus ;
            dm&j.x&i=(m&j-m&j.r&i)**2 ;
        %end ;
    %end ;
%mend ;
%it ;

%macro final ;

```

```
%do j=1 %to &numvar ;  
  data last ;  
    set all ;  
    sm&j=sum(of dm&j.x1-dm&j.x42) ;  
    sem&j=sqrt(sm&j) ;  
proc print ;  
  var m&j sem&j ;  
run ;  
  
%end ;  
  
%mend ;  
%final ;  
  
%mend ;  
%jackmean(dept1,sexf,r4mw,r4nc,r4s,42,5) ;
```

Appendix Program 2

```
***Jackknife for OLS regression, written by Pat Berglund during winter 97**** ;
***Uses National Comorbidity Survey Data, Dr. Ronald Kessler, director***** ;

libname max 'c:\data' ;
options symbolgen mprint ;

%macro jackreg(depvar,x1,x2,numclus,numpred) ;
data jrr ;
    set max.ncsdxdm2;
    p1=&x1 ;
    p2=&x2 ;
%macro wgtcal ;
    %do i=1 %to &numclus ;
        partwt&i=p1fwt ;
        if str=&i and secu=1 then partwt&i=partwt&i*2 ;
        if str=&i and secu=2 then partwt&i=0 ;
    %end ;

%mend ;
%wgtcal ;
*create the base predictors ;
proc reg data=jrr outest=parms ;
    model &depvar=p1-p2 ;
    weight p1fwt ;
run ;

%macro reps ;
    %do i=1 %to &numclus ;
        proc reg data=jrr outest=parms&i noprint ;
            model &depvar=p1-p2 ;
            weight partwt&i ;
            run ;
            data parms&i ;
            set parms&i ;
            %do j=1 %to &numpred ;
                rename p&j=p&j.x&i ;
            %end ;
        %end ;
    %mend ;
%reps ;

data rep ;
    merge
        parms
        parms1 parms2 parms3 parms4 parms5 parms6 parms7 parms8 parms9 parms10
        parms11 parms12 parms13 parms14 parms15 parms16 parms17 parms18 parms19 parms20
        parms21 parms22 parms23 parms24 parms25 parms26 parms27 parms28 parms29 parms30
        parms31 parms32 parms33 parms34 parms35 parms36 parms37 parms38 parms39 parms40
        parms41 parms42 ;
run ;
data all ;
    set rep ;
%macro it ;
    %do j=1 %to &numpred ;
        %do i=1 %to &numclus ;
            dp&j.x&i=(p&j-p&j.x&i)**2 ;
        %end ;
    %end ;
%mend ;
%it ;

%macro final ;
    %do j=1 %to &numpred ;
        data last ;
        set all ;
```

```
sp&j=sum(of dp&j.x1-dp&j.x42);  
sep&j=sqrt(sp&j);  
proc print ;  
var p&j sep&j ;  
run ;  
%end ;  
  
%mend ;  
%final ;  
  
%mend ;  
%jackreg(income,educ,age,42,2);
```